Wikipedia Search Engine using various Embedding Techniques
(Using BERT with user feedback)

Authors: Abhishek Nigam, Dhananjay Bagul, Rohit Kesarwani, Shambhavi Sharma, Sushil Kumar, Wilson Radadia

- **What is the problem identified in the project?**
  The problem identified in the project is that traditional search engines rely on keyword matching, which can result in incomplete and irrelevant results. The motivation behind building a BERT-based Wikipedia search engine with user feedback is to improve accuracy and relevance for users. BERT is a state of the art of natural language processing model that can generate text embeddings, allowing for a more sophisticated matching and ranking algorithm. Incorporating user feedback into search engines will enable users to provide explicit feedback on the relevance of search results which can be used further to filter the search algorithm and improve outcomes. By BERT-based Wikipedia search engine with user feedback, we can provide a valuable tool for people searching in Wikipedia.
- **Why is this problem important?**
  Projects like the BERT-based Wikipedia search engine with user feedback are significant to design because almost every user goes through a pervasive problem while searching for information online: "the lack of relevance and accuracy of search results".
  Natural language processing models like BERT can help improve search results accuracy by going through the hints given by the language and generating more accurate embeddings for text.
- **Is there any related work?**
  - "On Improving Wikipedia Search using Article Quality" by Meiqun Hu et al. which proposed a framework that re-ranks search results based on article quality based on two development quality measurement models, namely Basic and PeerReview, based on co-authoring data gathered from articles' edit history.
  - "Leveraging BERT for Extractive Text Summarization on Lectures" by Derek Miller which proposes a mechanism for extractive summarization through the use of K-Means clustering algorithm and the BERT model for text embeddings to find the closest sentences.
  - "A survey on word embedding techniques and semantic similarity for paraphrase identification" by Divesh R. Kubal*, Anant V. Nimkar aims to provide an analysis of existing similarity metrics, statistical machine translation metrics and includes different word embedding techniques, stepwise derivation of its learning module.
  - “NLP based Intelligent News Search Engine using Information Extraction from e-Newspapers” by Monisha Kanakaraj, Sowmya Kamath S present a personalized news search engine that focuses on building a repository of news articles by applying

efficient extraction of text information using TF-IDF from a web news page from varied e-news portals.
    - “Sarcasm Detection in Tweets with BERT and GloVe Embeddings” by Akshay Khatri, et al. proposed a machine learning technique with BERT and GloVe embeddings to detect sarcasm in tweets.
- **How different is your idea from theirs?**
    - We are incorporating user feedback which helps the search engine learn and adapt the user preferences and needs over time. It enhances user engagement with the search engine.
    - Using BERT, the search engine can better understand the meaning behind a user's query and provide more accurate search results.
- **What techniques/algorithms will you use/develop to solve the problem?**
    - Preprocess the Wikipedia Dataset: The first step is to preprocess the Wikipedia dataset by cleaning and tokenizing the text. You will also want to generate embeddings for each Wikipedia article using a pre-trained BERT model.
    - User Interface(UI): Develop a user interface that allows users to enter a search query and view the search results. Each search result should include the article's title, a brief summary, and a rating system that allows users to rate the result's relevance.
    - Search Algorithm: When a user enters a query, the search algorithm should use a BERT-based model to generate embeddings for the question. These embeddings can then be compared to the embeddings of each Wikipedia article to calculate a similarity score. The search algorithm should return the top N results based on their similarity score.
    - User feedback: After the search results are displayed, users should be able to rate the relevance of each result using a rating system. These ratings can improve future search results by adjusting the search algorithm accordingly. For example, if users consistently rate a particular result highly, the search engine could boost the ranking of similar results in future searches.
    - Re-ranking of search results: After step-4, the search algorithm should be re-run using the new ratings to re-rank the search results. This process can be repeated iteratively to improve the accuracy of the search results over time.
    - Deployment and monitoring: Once the search engine is developed, it should be deployed to a production environment and monitored for performance and user feedback.
- **How will you evaluate your work?**

  We can evaluate the performance of the BERT-based Wikipedia search engine in several ways. Following are some possible evaluation metrics:
    - Precision and Recall: These metrics measure the accuracy and completeness of the search engine result. Precision is the fraction of retrieved documents relevant to the query, while recall is the fraction of relevant documents retrieved.

- Mean Average Precision(MAP): This metric measures the average precision of the search engine's results over a set of queries. It considers the relevance of each document and the position at which it appears in the ranking.

- **List your potential contributions to this work.**
  Our contributions will be to add a personalized search experience to the user such that it can increase the user engagement with the search engine and increase the likelihood of repeat usage. Our work can serve as a potential commercial application for improving product search on e-commerce websites. So overall, a well-designed and effective Wikipedia search engine using BERT and incorporating user feedback has the potential to provide significant benefits to users and businesses alike.

  Individual Contribution.
  Abhishek : Re-ranking of search results and deployment and monitoring
  Dhananjay : Re-ranking of search results and deployment and monitoring
  Rohit : Idea formation, Search algorithm and user feedback implementation
  Shambhavi : Preprocessing and UI interface
  Sushil : Idea formation, Search algorithm and user feedback implementation
  Wilson : Preprocessing and UI interface

- **References**

  1. Hu, Meiqun, et al. "On improving wikipedia search using article quality." Proceedings of the 9th annual ACM international workshop on Web information and data management. 2007.
  2. Miller, Derek. "Leveraging BERT for extractive text summarization on lectures." arXiv preprint arXiv:1906.04165 (2019).
  3. Kubal, Divesh R., and Anant V. Nimkar. "A survey on word embedding techniques and semantic similarity for paraphrase identification." International Journal of Computational Systems Engineering 5.1 (2019): 36-52.
  4. Kanakaraj, Monisha, and S. Sowmya Kamath. "NLP based intelligent news search engine using information extraction from e-newspapers." 2014 IEEE International Conference on Computational Intelligence and Computing Research. IEEE, 2014.
  5. Khatri, Akshay. "Sarcasm detection in tweets with BERT and GloVe embeddings." arXiv preprint arXiv:2006.11512 (2020).